

# Streamlining Report Reusability in SAS Viya: A REST API Approach

SM07

Mary Dolegowski  
David Weik



# Meet The Authors

## Streamlining Report Reusability in SAS Viya: A REST API Approach



**Mary Dolegowski**

Company: SAS Institute

City: Cary, NC, USA

Email:

[Mary.Dolegowski@sas.com](mailto:Mary.Dolegowski@sas.com)



**David Weik**

Company: SAS Institute

City: Heidelberg, Germany

Email:

[David.Weik@sas.com](mailto:David.Weik@sas.com)

# Agenda

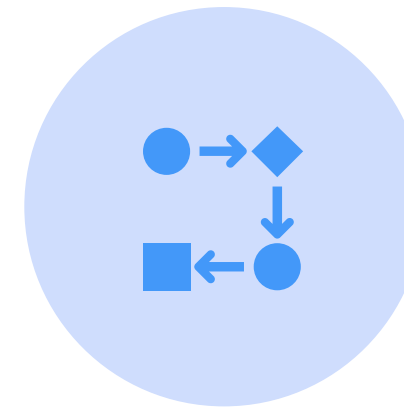
## Streamlining Report Reusability in SAS Viya: A REST API Approach



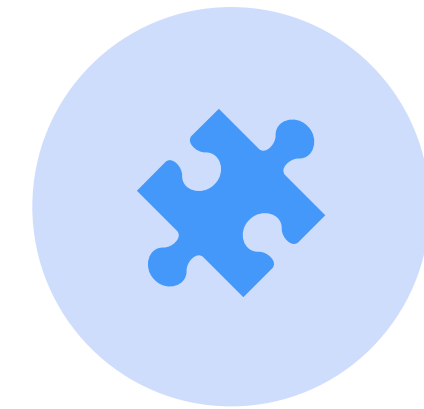
WHY



SAS VIYA APIS



API PROCESS FLOW



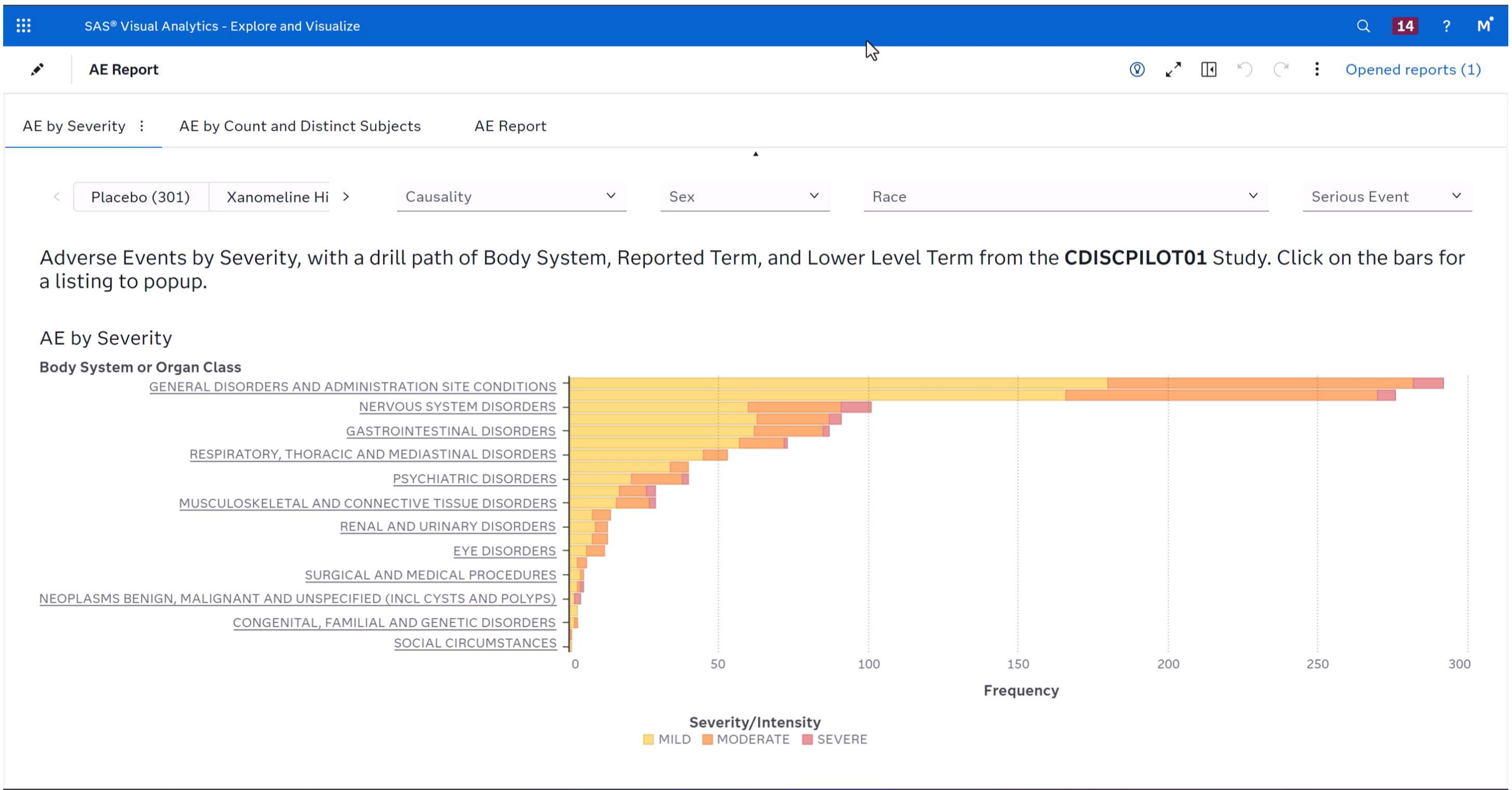
PUTTING IT ALL  
TOGETHER

# Why



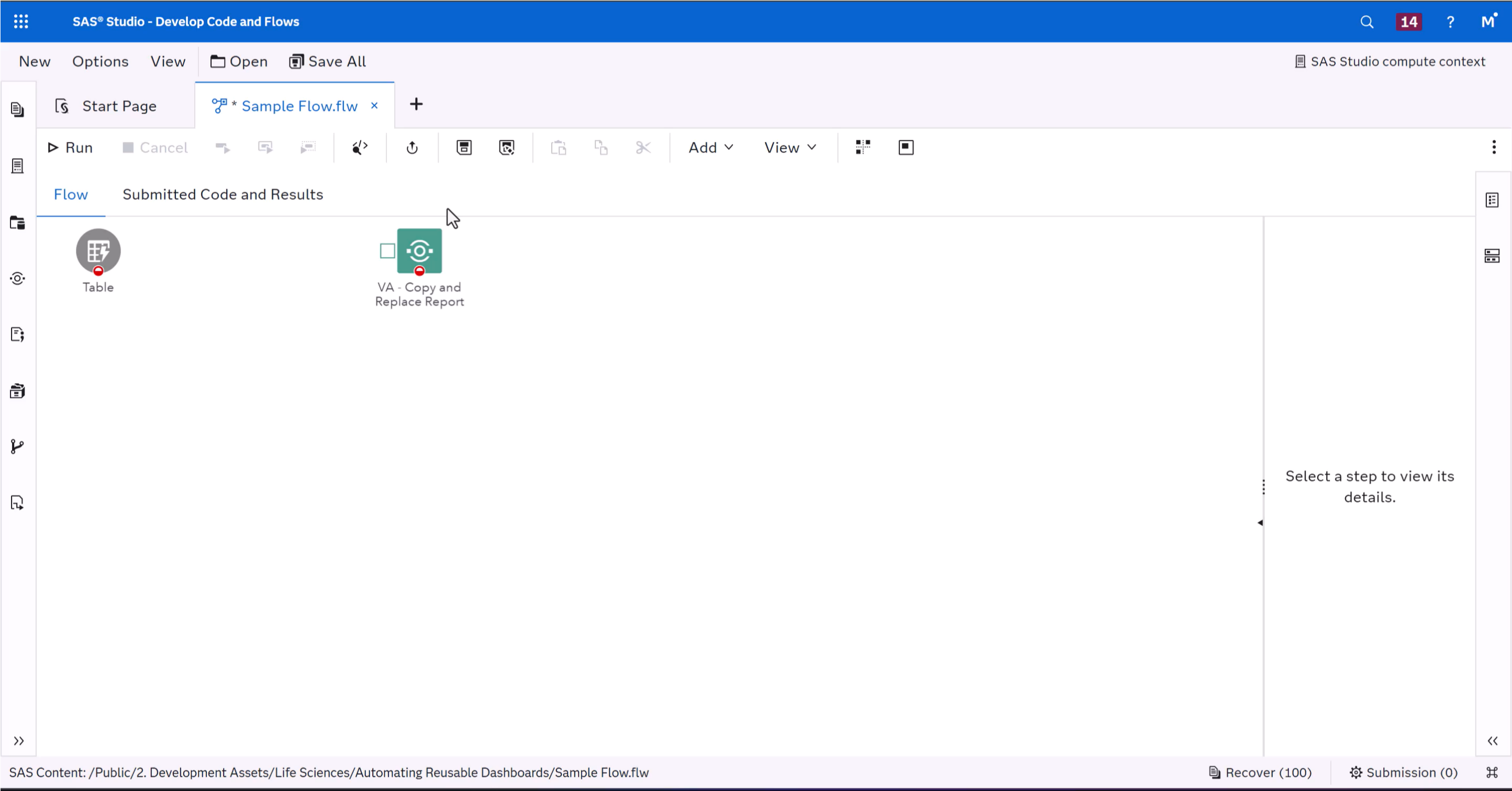
# Current Process

## Why



# Updated Process

Why



# SAS Viya APIs



# SAS Viya APIs

[illegible]

# Sample SAS API

## SAS Viya APIs

- \* API call to copy the original report and save it in the new folder;
- \* <https://developer.sas.com/rest-apis/visualAnalytics/updateReportCopy>;

**proc http**

`url = "&_crr_viyaHost./visualAnalytics/reports/&_crr_original_report_id./copy"` URL specifying which API

`method = 'Put'`

`in = _crr_in` The in file if needed – in this case it specifies details around the report to be copied

`out = _crr_out` Specifies the out file – in this case we need the detail in this out file for the replace process

`oauth_bearer = sas_services;` Special SAS step – Viya API calls within SAS Viya do not need the authentication handshake as users have already been authenticated when they log in

`headers`

```
'Accept' = 'application/json,  
application/vnd.sas.report.operations.results+json,  
application/vnd.sas.report.operations.error+json, application/vnd.sas.error+json'  
  
'Content-Type' = 'application/json';
```

**run;**

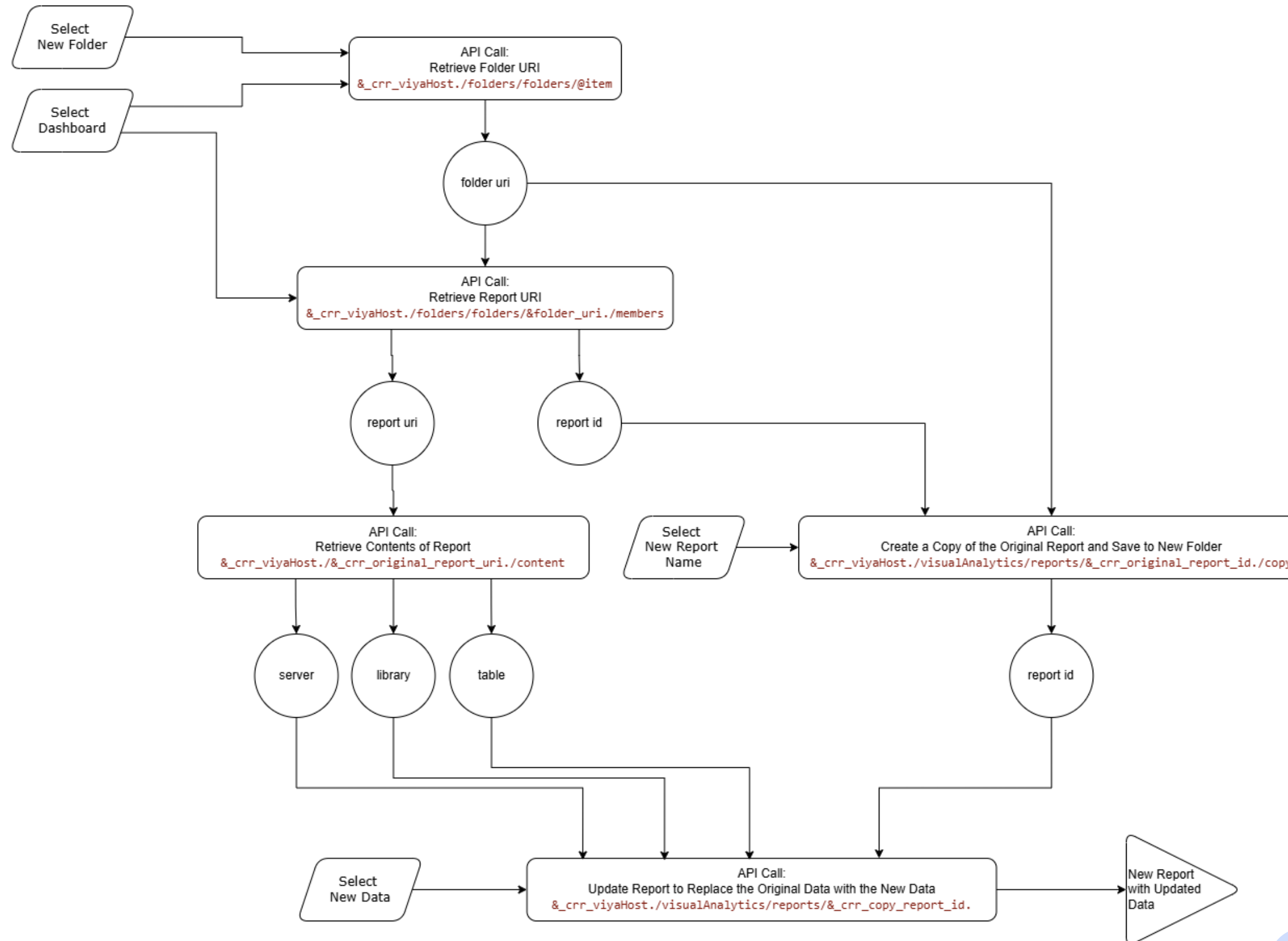


# API Process Flow



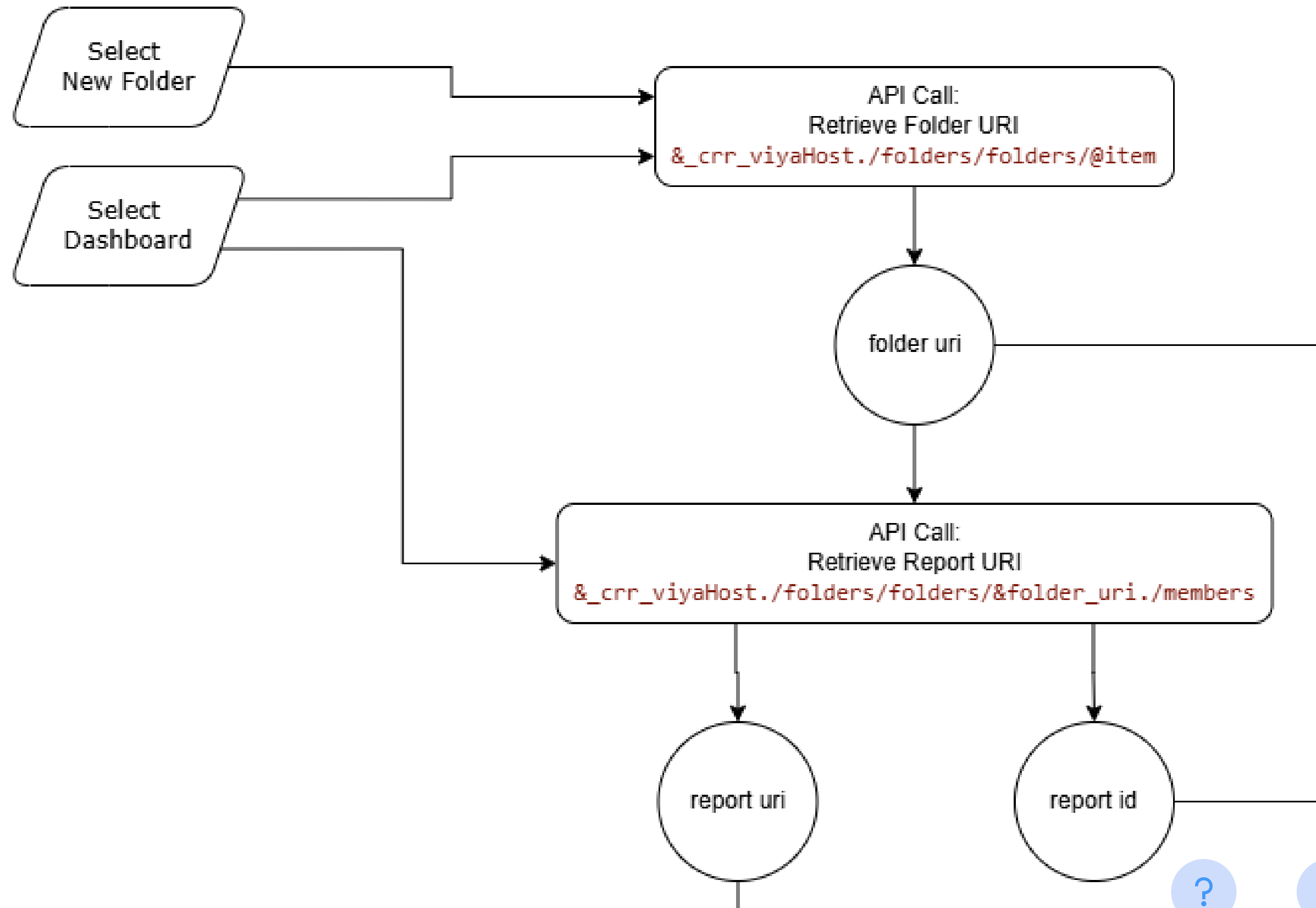
# Process Flow

## API Process Flow



# Process Flow

## API Process Flow

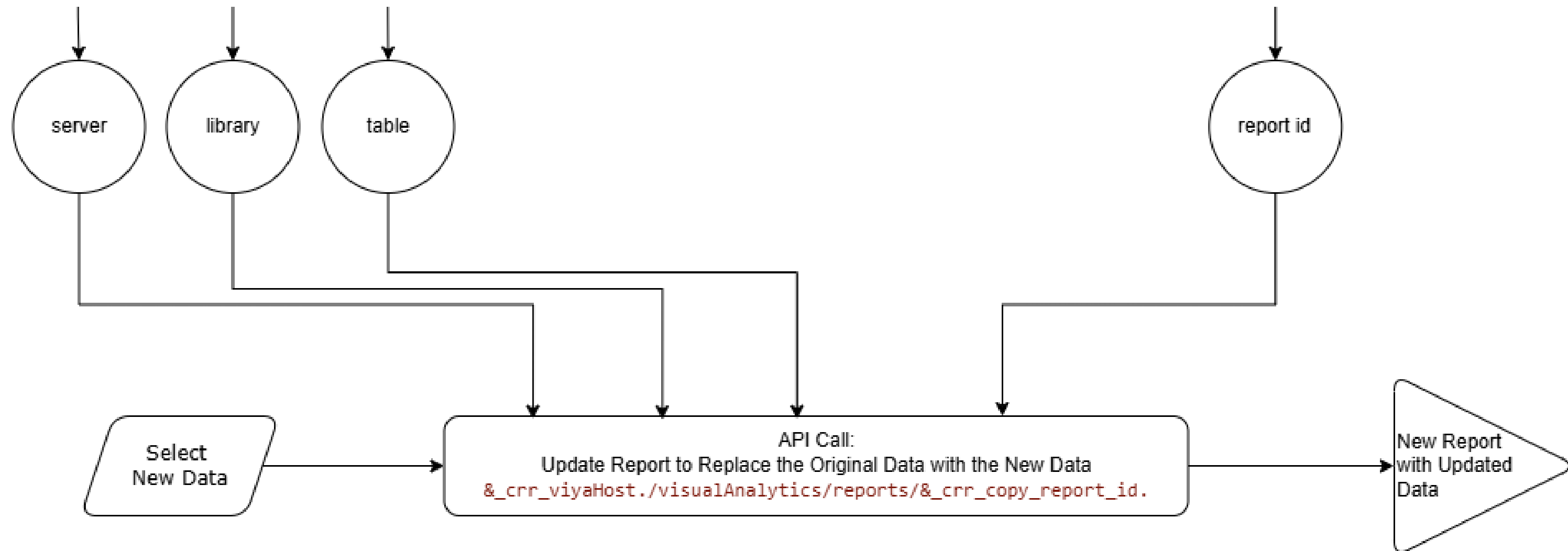


# API Process Flow



# Process Flow

## API Process Flow



# Sample JSON

## API Process Flow

\* Create the request body for the copy API call;

**proc cas;**

```
set stdJSON;
resp_file = readPath(&_crr_out_path.);
resp_string = json2CASL(resp_file);

do item over resp_string.items;
    report_uri = item.uri;
end;

report_id = scan(report_uri, 3, '/');
symputx('_crr_original_report_id', report_id, 'G');
symputx('_crr_original_report_uri', report_uri, 'G');
```

```
in_data.resultFolder = "&_crr_new_folder_uri.";
in_data.resultReportName = "&_crr_name_update.";
in_data.resultNameConflict = 'replace';
```

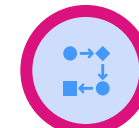
```
in_json = casl2JSON(in_data);
in_json = compbl(in_json);
```

```
file outfile &_crr_in_path.;
print in_json;
```

**run;**

Reading in JSON data from the out file of a previous API call and assigning the values to macro variables for future use

Creating data in a dictionary structure to then be turned into JSON for the next steps in file



# Putting it All Together



# Custom Step and Macros

## Putting it All Together

SAS® Studio - Develop Code and Flows

New Options View Open Save All

Start Page VA - Copy and Replace Report x +

Run Cancel Details

Options About

This custom step will take an existing SAS Visual Analytics report and either copy the report and keep the original data, or copy the report and replace the original data with a new data set.

Right now this only works for a single dataset.

Select a report: \*

Select one:

☐ Copy report and keep original data

☒ Copy report and change source data

Select data replacement - has to be a CAS table:

If blank the report will be saved to the same folder as the original report.

(Optional) Select new report location within SAS Content:

If blank " - Copy" will be appended to the reports name.

(Optional) New report name:

SAS Content: /Public/1. Gold Assets/Life Scie... Recover (100) Submission (0)

SAS® Studio - Develop Code and Flows

New Options View Open Save All SAS Studio compute context

Start Page Sample Flow.flw VA - Copy and Replace Report.step x +

Design JSON Program

```
70
71 * Retrieve the folder ID for the new report location;
72 * https://developer.sas.com/rest-apis/folders/getFolderItem;
73 proc http
74   url = "&_crr_viyaHost./folders/folders/@item?path=&_crr_new_report_loc."
75   method = 'Get'
76   out = _crr_out
77   oauth_bearer = sas_services;
78   headers
79     'Accept' = 'application/json, application/vnd.sas.collection+json, application/vnd.sas.error+json'
80 run;
81
82 * Check response;
83 data _null_;
84   if &SYS_PROCHTTP_STATUS_CODE. ne 200 then do;
85     putLog 'ERROR: Request was not successfull, please make sure you have selected a folder in SAS Co
86     abort;
87   end;
88 run;
89
90 * Parse the folder ID from the response;
91 proc cas;
92   set stdJSON;
93   resp_file = readPath(&_crr_out_path.);
94   resp_string = json2CASL(resp_file);
95   new_folder_uri = '/folders/folders/' || resp_string.id;
96   symputx('_crr_new_folder_uri', new_folder_uri, 'G');
97 run;
98
99 * Retrieve the folder ID for the original report location;
100 * https://developer.sas.com/rest-apis/folders/getFolderItem;
```

SAS Content: /Public/1. Gold Assets/Life Sciences/VA - Cope and Replace Report/VA - Copy and... Recover (100) Submission (0)

Custom Steps are essentially macros with a GUI interface.

Using the code from one to create the other is quick and easy.

# Future Iterations

## Putting it All Together

- Expand the process to include more than one dataset
- Quick look up set of commonly used reports



# Questions

